

I. Spécification des besoins

I.1 Identification des acteurs

Les principaux acteurs qui interagissent avec le système sont :

- **Internaute**: un acteur qui peut créer un compte administrateur ou autres utilisateurs
- **Administrateur (Banque du sang)** : Un acteur qui est chargé gérer les utilisateurs (ajouter, modifier, supprimer et lister)
- **Médecin** : Il est chargé de vérifier et certifier l'aptitude des donneurs à pouvoir donner leur sang.
- **Infirmier** : Il est chargé de faire toutes les actions allant des prélèvements des produits sanguins aux validations des dons de sang

I.2 Identification des besoins fonctionnels

1/ Module Accès

Les acteurs Médecin, Infirmier et Administrateur ont le besoin commun suivant:

- **S'authentifier**: L'utilisateur peut s'authentifier et accéder à son compte.
- **Gérer Profil** : L' utilisateur peut consulter son profil et modifier ses informations. Il a aussi la possibilité de supprimer son compte.

2/ Module Approvisionnement

A. Besoins fonctionnels de l'acteur «Internaute»

- **S'inscrire**: Un internaute peut créer son propre compte

B. Besoins fonctionnels de l'acteur « Administrateur»

- **Gérer les utilisateurs (lister les utilisateurs et supprimer)**: L'administrateur peut consulter les différents utilisateurs, lister ou supprimer un utilisateur.
- **Gérer les donneurs (lister les donneurs, modifier et supprimer)**
- **Lister les dons** : L'administrateur a la possibilité de consulter les dons et de suivre leur état.

C. Besoins fonctionnels de l'acteur « Médecin »

- **Gérer les donneurs (lister les donneurs, ajouter, modifier, supprimer) :**
Le Médecin peut consulter les différentes informations concernant les donneurs: ajouter, lister, modifier et supprimer :
- **Gérer les dons (lister les dons, ajouter, modifier, supprimer) :** consulter les informations des donneurs: Valider le don

D. Besoins fonctionnels de l'acteur « Infirmier »

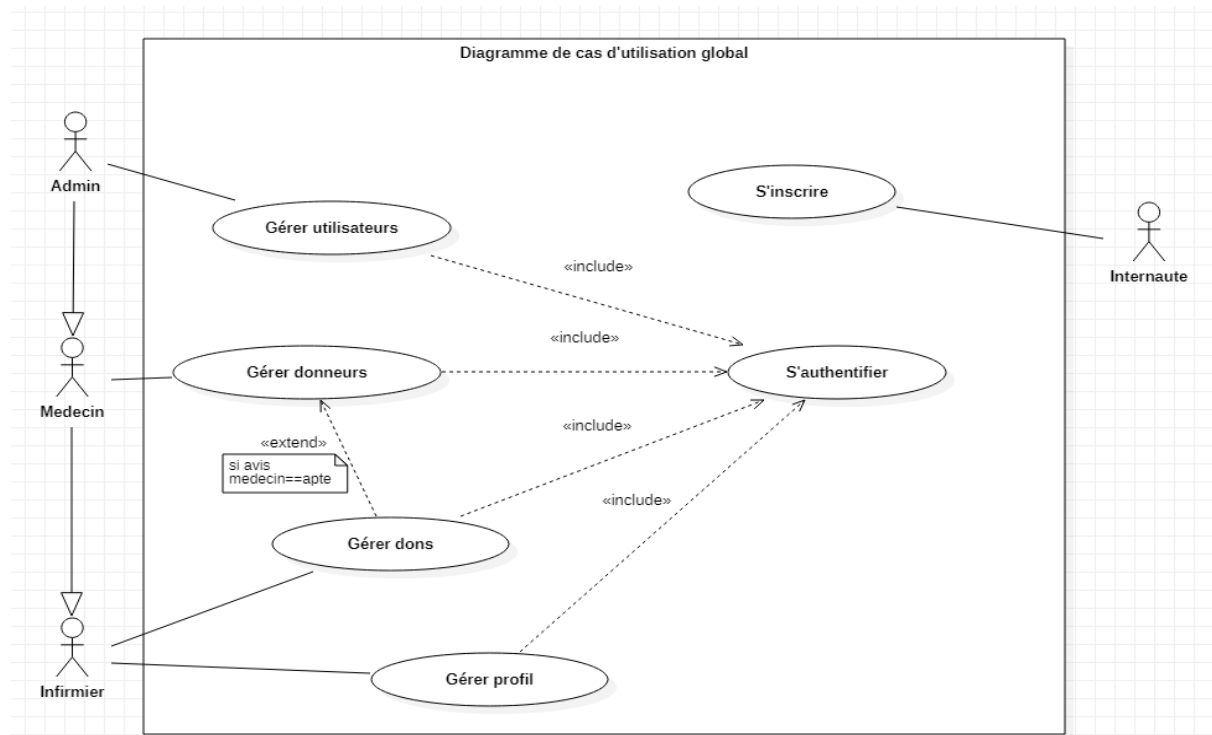
- **Gérer les donneurs (Lister et modifier)**
- **Gérer les dons (modifier, lister les dons) :** consulter les informations des donneurs ainsi que l'avis du médecin: faire prélèvement, si le don est validé

I.2 Identification des besoins non fonctionnels

- **Ergonomie :** L'application doit être facile à utiliser en offrant une interface conviviale et simple.
- **Rapidité :** L'application doit permettre un temps de réponse raisonnable avec des traitements optimisés.
- **Sécurité :** L'application doit garantir la confidentialité et la sécurité des données personnelles des utilisateurs. De plus, chaque utilisateur accédant à son espace doit être protégé par un mot de passe et des droits d'accès.
- **Maintenabilité :** L'application doit fournir des solutions faciles à maintenir avec un code lisible, compréhensible et bien commenté.

II. Structure et découpage de projet

Diagramme de cas d'utilisation global



RAPPORT SGBD PFA

SYSTÈME DE GESTION DE COLLECTE DE SANG

IMEN FREDJ BI 1

NADA ARFAOUI BI 1

on a essayé d'appliquer toutes les notions qu'on a vu dans notre cours:

1)VUE

une vue constituant une restriction de la table donneur aux donneurs du type de sang A+(ce type de sang est rare, donc on voulait savoir les id et les adresses des donneurs portant ce type de sang pour les contacter):

```
CREATE VIEW typeApos AS SELECT id_donneur, adresse FROM donneur WHERE type_sang = "A+";
```

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0350 seconde(s).)

```
CREATE VIEW typeApos AS SELECT id_donneur, adresse FROM donneur WHERE type_sang = "A+";
```

[Éditer en ligne] [Éditer] [Créer le code source PHP]

La sélection courante ne contient pas de colonne unique. Les grilles d'édition, les liens Éditer, Copier et Supprimer peuvent provoquer un résultat indésirable.

✓ Affichage des lignes 0 - 0 (total de 1, traitement en 0,0015 seconde(s).)

```
SELECT * FROM `typeapos`
```

☐ Profilage [Éditer en ligne] [Éditer] [Expliquer SQL] [Créer le code source PHP] [Actualiser]

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table

+ Options

id_donneur adresse

☐ Éditer ☐ Copier ☐ Supprimer 14450421 ben arouss

☐ Tout cocher Avec la sélection : ☐ Éditer ☐ Copier ☐ Supprimer ☐ Exporter

☐ Tout afficher | Nombre de lignes : 25 | Filtrer les lignes: Chercher dans cette table

2)indexe

On consulte fréquemment les médecins dans la table médecin qui travaillent dans les centres, ils sont plus disponibles, alors on veut afficher les médecins qui travaillent dans les centres

*On crée un index sur la table médecin sur la colonne centre.

```
CREATE INDEX centre_medecin ON medecin(centre);
```

✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0228 seconde(s).)

```
CREATE INDEX centre_medecin ON medecin(centre);
```

[Éditer en ligne] [Éditer] [Créer le code source PHP]

Action	Nom de l'index	Type	Unique	Compressé	Colonne	Cardinalité	Interclassement	Null	Commentaire
<input type="checkbox"/> Éditer <input type="checkbox"/> Renommer <input type="checkbox"/> Supprimer	PRIMARY	BTREE	Oui	Non	id_medecin	5	A	Non	
<input type="checkbox"/> Éditer <input type="checkbox"/> Renommer <input type="checkbox"/> Supprimer	centre_medecin	BTREE	Non	Non	centre	5	A	Non	

Créer un index sur 1 colonnes Exécuter

3)Création d'une séquence:

CREATE SEQUENCE identifiant_don MINVALUE 100 MAXVALUE 500 START WITH 103 INCREMENT BY 1 ;

Afficher la zone SQL

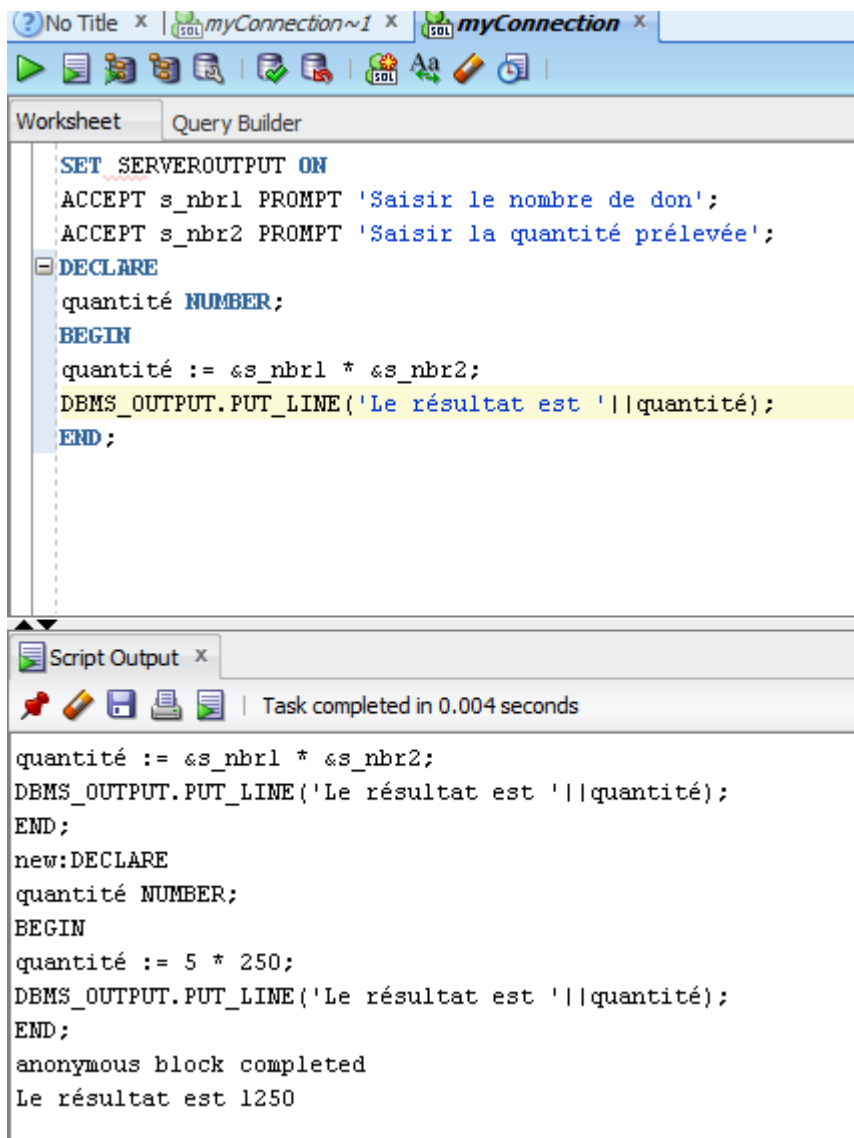
✓ MySQL a retourné un résultat vide (c'est à dire aucune ligne). (traitement en 0,0281 seconde(s).)

```
CREATE SEQUENCE identifiant_don MINVALUE 100 MAXVALUE 500 START WITH 103 INCREMENT BY 1;
```

[Éditer en ligne] [Éditer] [Créer le code source PHP]

5)Requête d'interrogation et de modification des tables de BD

une fonction pour les infirmiers, qui retourne la quantité du sang prélevée en l'affectant le nombre des donneurs



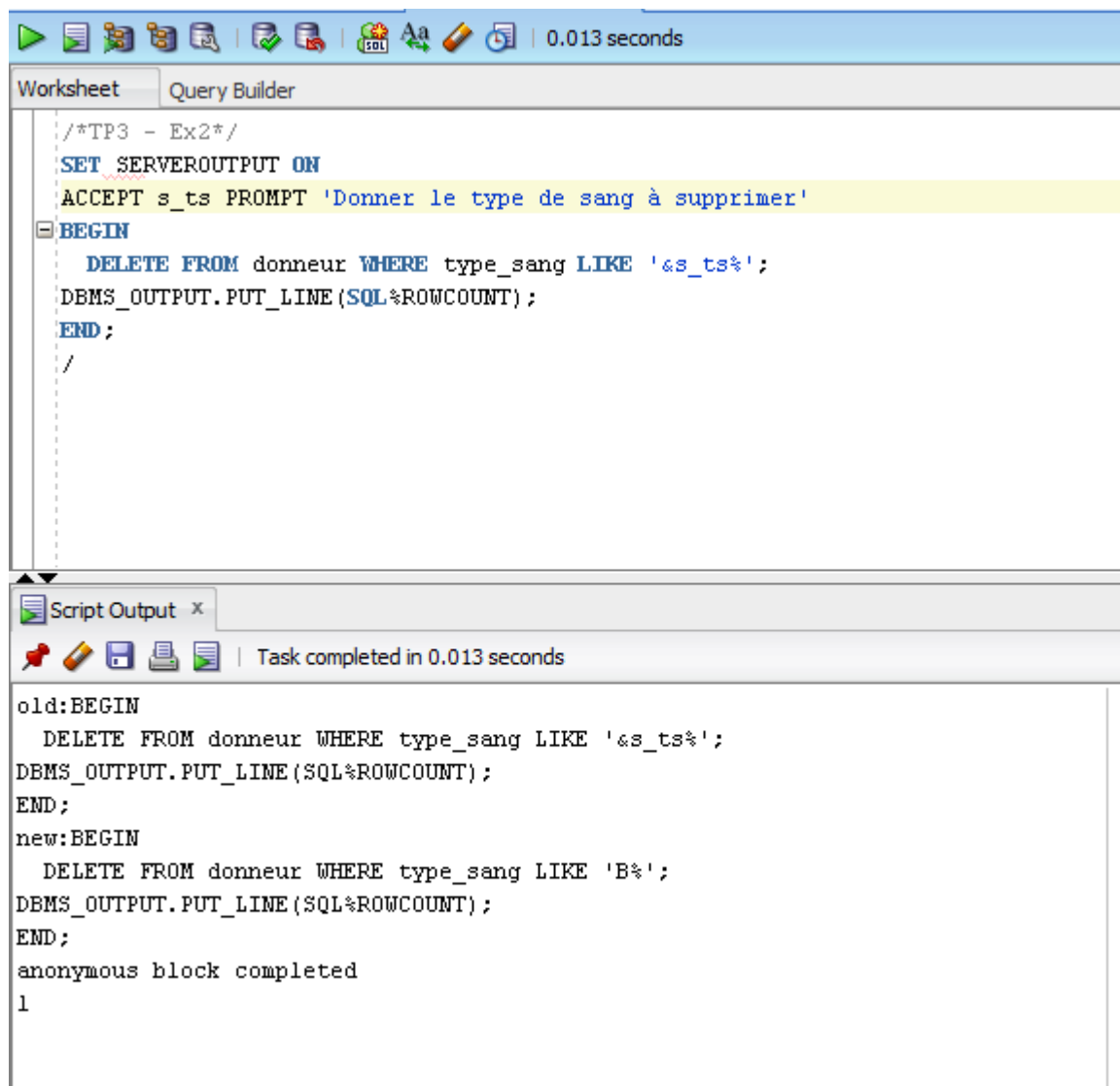
The screenshot shows a SQL IDE window with two tabs: 'Worksheet' and 'Query Builder'. The 'Worksheet' tab is active, displaying a PL/SQL script. The script defines a variable 'quantité' of type 'NUMBER', calculates its value as the product of two input variables 's_nbr1' and 's_nbr2', and outputs the result using 'DBMS_OUTPUT.PUT_LINE'. The script is executed, and the 'Script Output' window at the bottom shows the execution details, including the calculation of 'quantité' as 1250 (5 * 250) and the final output message 'Le résultat est 1250'.

```
SET SERVEROUTPUT ON
ACCEPT s_nbr1 PROMPT 'Saisir le nombre de don';
ACCEPT s_nbr2 PROMPT 'Saisir la quantité prélevée';
DECLARE
quantité NUMBER;
BEGIN
quantité := &s_nbr1 * &s_nbr2;
DBMS_OUTPUT.PUT_LINE('Le résultat est '||quantité);
END;
```

Script Output x

Task completed in 0.004 seconds

```
quantité := &s_nbr1 * &s_nbr2;
DBMS_OUTPUT.PUT_LINE('Le résultat est '||quantité);
END;
new:DECLARE
quantité NUMBER;
BEGIN
quantité := 5 * 250;
DBMS_OUTPUT.PUT_LINE('Le résultat est '||quantité);
END;
anonymous block completed
Le résultat est 1250
```



The screenshot displays the Oracle SQL Developer environment. The top toolbar includes icons for running, saving, and other database operations, with a timer showing 0.013 seconds. The 'Worksheet' tab is active, showing a SQL script. The 'Script Output' window at the bottom shows the execution results.

```
/*TP3 - Ex2*/
SET SERVEROUTPUT ON
ACCEPT s_ts PROMPT 'Donner le type de sang à supprimer'
BEGIN
    DELETE FROM donneur WHERE type_sang LIKE '&s_ts%';
    DBMS_OUTPUT.PUT_LINE(SQL%ROWCOUNT);
END;
/
```

The Script Output window shows the following execution results:

```
old:BEGIN
  DELETE FROM donneur WHERE type_sang LIKE '&s_ts%';
DBMS_OUTPUT.PUT_LINE(SQL%ROWCOUNT);
END;
new:BEGIN
  DELETE FROM donneur WHERE type_sang LIKE 'B%';
DBMS_OUTPUT.PUT_LINE(SQL%ROWCOUNT);
END;
anonymous block completed
1
```

une fonction qui supprime un donneur qui porte un type de sang spécifique ,elle aide l'admin à gérer les types du sang rare dans la base

la base de donnée avant la suppression des donneur qui portent B-

ORACLE Database Express Edition

Utilisateur: USERN

Page d'accueil > SQL > Commandes SQL

☒ Validation automatique Afficher 10

Enregistrer Exécuter

SELECT * FROM donneur;

Résultats Expliquer Décrire SQL enregistré Historique

ID_DONNEUR	NOM	PRENOM	AGE	SEXE	ADRESSE	EMAIL	NUMERO	TYPE_SANG	POIDS	TAILLE	DERNIER_DON	M...
14050231	flouhi	salma	38	f	tunis	salma.flouhi@gmail.com	28478524	B-	85	170	10/11/18	sai
14425876	madani	ahmed	43	m	ariana	ahmed.madani@gmail.com	99124568	O-	86	161	12/02/21	ter
14450111	mamissi	ghofrane	20	f	manouba	ghofrane.mamissi@gmail.com	28557132	A+	61	155	12/01/20	sai
14450123	riahi	amine	35	m	tunis	amine.riahi@gmail.com	98954157	O+	60	160	22/01/20	sai
14450421	hamdi	oifa	20	f	ben arouss	oifa.hamdi@gmail.com	99124578	A+	55	155	26/04/22	sai
14450422	jouini	hedi	45	m	ouina	hedi.jouini@gmail.com	99124478	O+	72	166	20/03/20	sai

6 lignes renvoyées en 0.00 secondes Export CSV

Application Express 2.1.0.00.39

la base de donnée après la suppression des donneur qui portent B-

No Title x myConnection~1 x myConnection x MEDECIN x

0.02 seconds

myConnection

Worksheet Query Builder

SELECT * FROM donneur ;

Script Output x

Task completed in 0.02 seconds

NUMERO	TYPE_SANG	POIDS	TAILLE	DERNIER_DON	MALADIE	ID_DON
99124568	O-	86	161	12-FEB-21	tension	105
28557132	A+	61	155	12-JAN-20	saine	103
98954157	O+	60	160	22-JAN-20	sain	102
99124578	A+	55	155	26-APR-22	saine	100
99124478	O+	72	166	20-MAR-20	sain	101

5) curseur explicite

Écrire un bloc PL/SQL qui affiche pour chaque centre le nom du médecin, le nom du centre et le nombre de médecins qui y travaillent . (on a utilisé un seul curseur explicite)

SET SERVEROUTPUT ON

DECLARE

CURSOR emp_c IS SELECT * FROM medecin ORDER BY centre;

emp_r emp_c%ROWTYPE;

v_nbr NUMBER;

BEGIN

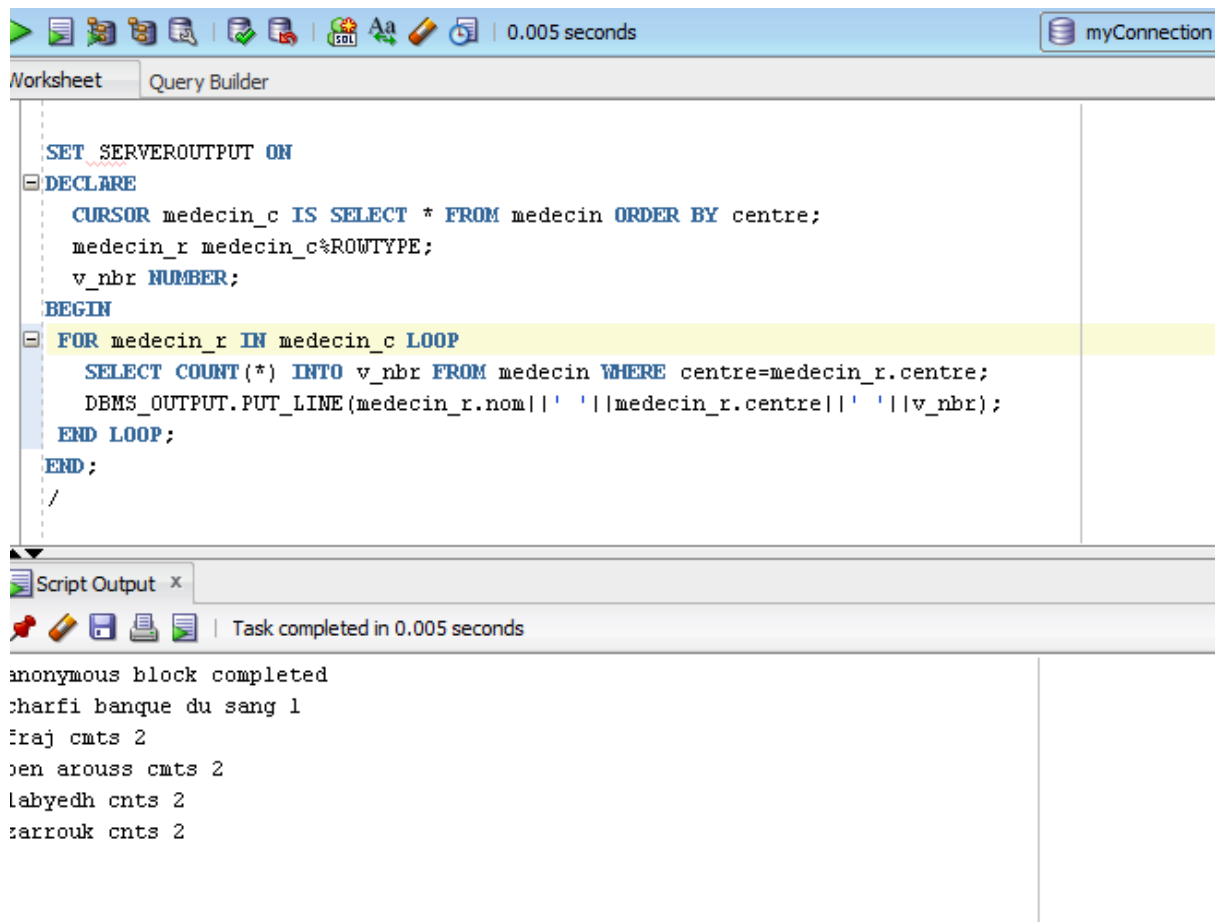
FOR emp_r IN emp_c LOOP

SELECT COUNT(*) INTO v_nbr FROM medecin WHERE centre=emp_r.centre;

DBMS_OUTPUT.PUT_LINE(emp_r.nom||' '||emp_r.centre||' '||v_nbr);

END LOOP;

END;



The screenshot shows a database IDE with a 'Query Builder' tab. The script in the editor is as follows:

```
SET SERVEROUTPUT ON
DECLARE
    CURSOR medecin_c IS SELECT * FROM medecin ORDER BY centre;
    medecin_r medecin_c%ROWTYPE;
    v_nbr NUMBER;
BEGIN
    FOR medecin_r IN medecin_c LOOP
        SELECT COUNT(*) INTO v_nbr FROM medecin WHERE centre=medecin_r.centre;
        DBMS_OUTPUT.PUT_LINE(medecin_r.nom||' '||medecin_r.centre||' '||v_nbr);
    END LOOP;
END;
/
```

Below the editor, the 'Script Output' window shows the results of the script execution:

```
anonymous block completed
charfi banque du sang 1
iraj cmts 2
ben arouss cmts 2
labyedh cmts 2
zarrouk cmts 2
```

The IDE interface includes a toolbar at the top with icons for running, saving, and other operations, and a status bar at the bottom indicating 'Task completed in 0.005 seconds'.

6)*les exeptions

on voulait écrire un programme qui nous permet de savoir si un donneur d'un nom défini existe ou non

```
SET SERVEROUTPUT ON
```

```
DECLARE
```

```
    v_don NUMBER;
```

```
BEGIN
```

```
    SELECT id_donneur INTO v_don FROM donneur WHERE nom='maha';
```

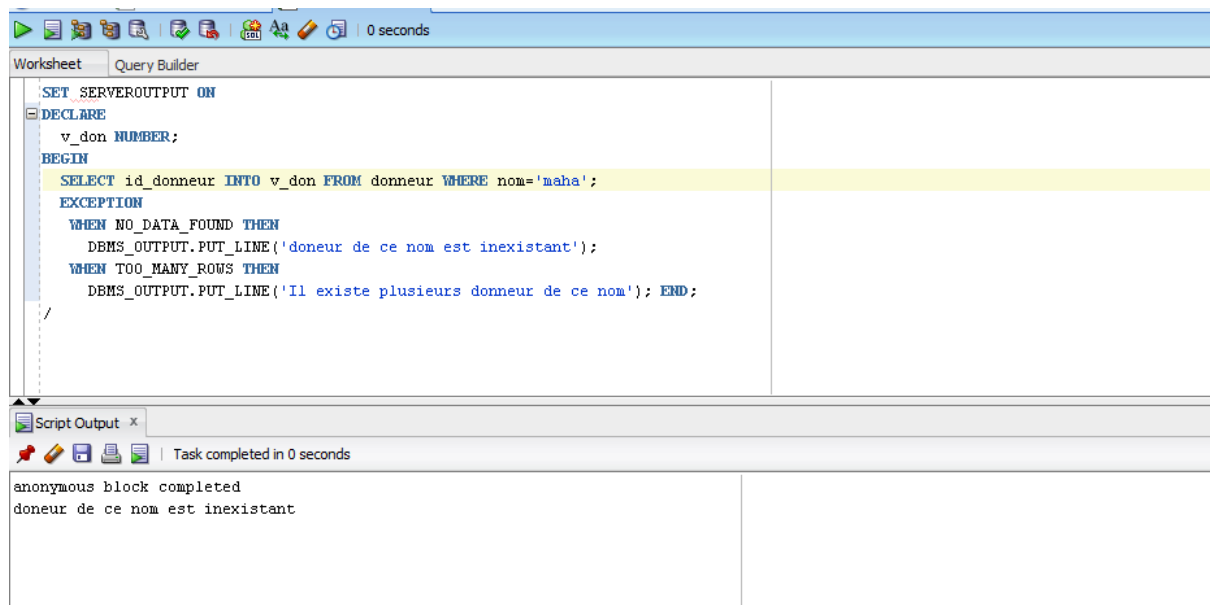
```
EXCEPTION
```

```
    WHEN NO_DATA_FOUND THEN
```

```
        DBMS_OUTPUT.PUT_LINE('doneur de ce nom est inexistant');
```

```
    WHEN TOO_MANY_ROWS THEN
```

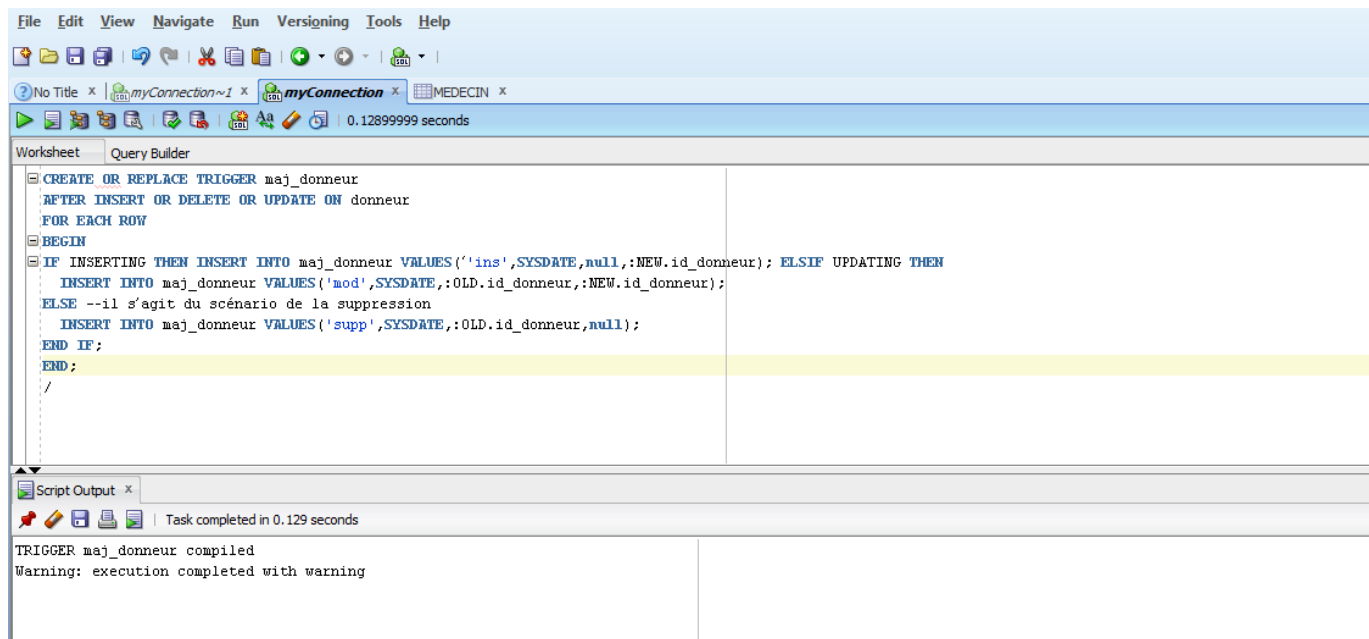
```
        DBMS_OUTPUT.PUT_LINE('Il existe plusieurs donneur de ce nom'); END;/
```

7)trigger

on veut créer un trigger sur la table donneur

```
CREATE OR REPLACE TRIGGER maj_donneur
AFTER INSERT OR DELETE OR UPDATE ON donneur
FOR EACH ROW
BEGIN
  IF INSERTING THEN      INSERT INTO maj_donneur VALUES('ins',SYSDATE,null,NEW.id_donneur); ELSIF
  UPDATING THEN
    INSERT INTO maj_donneur VALUES('mod',SYSDATE,OLD.id_donneur,NEW.id_donneur);
  ELSE --il s'agit du scénario de la suppression
    INSERT INTO maj_donneur VALUES('supp',SYSDATE,OLD.id_donneur,null);
  END IF;
END;
/
```



8)PROCEDURE

on veut afficher les noms et type de sang des donneurs qui ont fait un prélèvement pendant le dernier mois

```

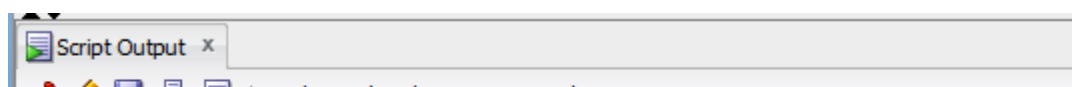
CREATE OR REPLACE PROCEDURE der_don (id_don IN NUMBER, p_nom OUT VARCHAR2, p_typesang
OUT NUMBER)
IS
BEGIN
SELECT nom, type_sang INTO p_nom, p_typesang FROM donneur
WHERE dernier_don BETWEEN 31-03-2022 AND 30-04-2022 AND ;
END der_don;

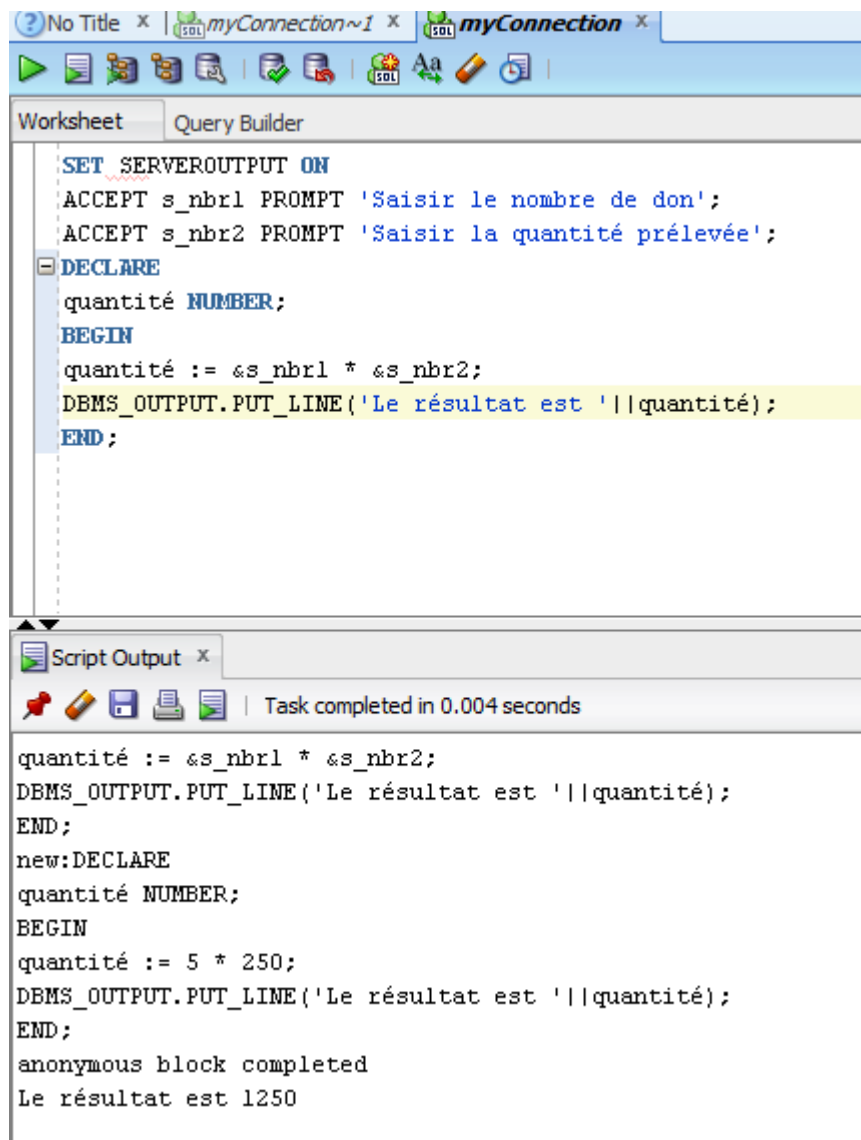
```

```

SET SERVEROUTPUT ON
DECLARE
id_don NUMBER;
v_nom VARCHAR2(20);
v_typesang NUMBER;
BEGIN
iddon:=14450427;
der_don(id_don,v_nom,v_typesang);
DBMS_OUTPUT.PUT_LINE('Le nom est '||v_nom||' de grade '||v_typesang);
END;
/

```





9) FONCTION

EN cas d'urgence , on a créé une fonction qui affiche le donneur situé un une ville donnée et qui porte un certain type de sang

```

CREATE OR REPLACE FUNCTION nb_donneur(p_type IN NUMBER, p_adresse IN NUMBER) RETURN
NUMBER
IS
v_nb NUMBER;
BEGIN
SELECT COUNT(*) INTO v_nb FROM emp
WHERE type_sang = p_type AND adresse = p_adresse;
RETURN v_nb;
END nb_donneur;

```

```

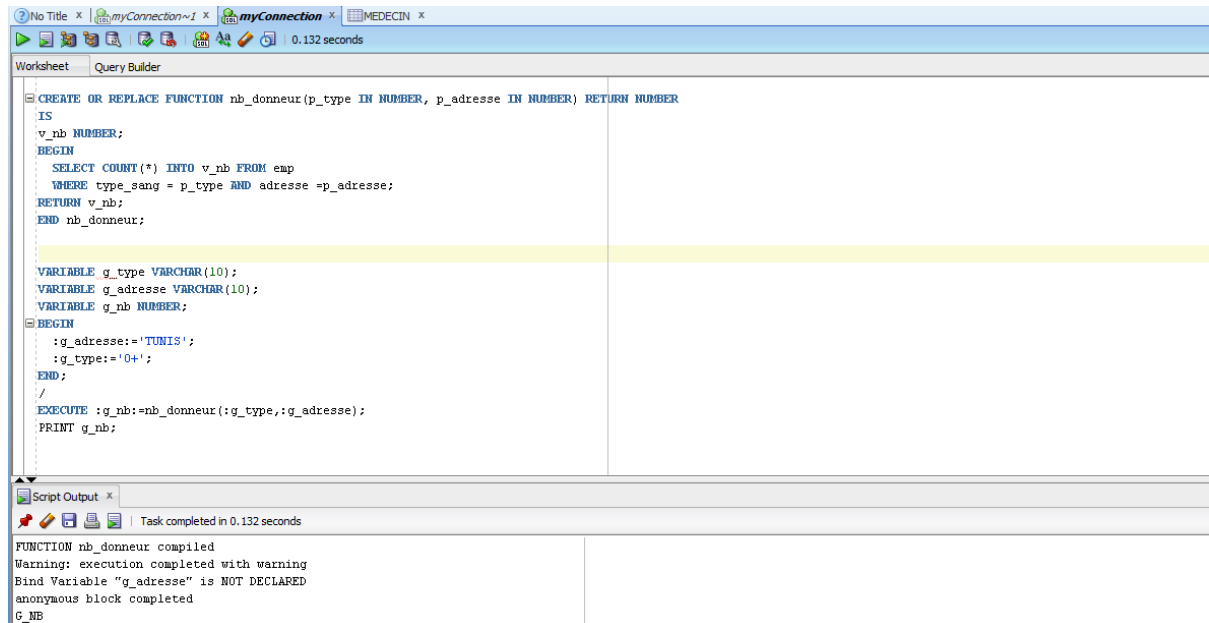
VARIABLE g_type VARCHAR(10);
VARIABLE g_adresse VARCHAR(10);

```

```

VARIABLE g_nb NUMBER;
BEGIN
  :g_adresse:='TUNIS';
  :g_type:='O+';
END;
/
EXECUTE :g_nb:=nb_donneur(:g_type,:g_adresse);
PRINT g_nb;

```



10)CURSEURS IMPLICITE

ceci est un code simple d'un curseur implicite

-- curseur_implicite1.sql

SET SERVEROUTPUT ON

BEGIN

-- modification des centres des infirmiers dont l'adresse est cnts en cmts

UPDATE infirmier

SET centre='cnts'

WHERE centre='cmts';

END;

/

```

0.023 seconds

Worksheet | Query Builder

-- curseur_implicitel.sql
SET SERVEROUTPUT ON

BEGIN
-- modification des centres des infermiers dont l'adresse est cnts en cmts
UPDATE infermier
SET centre='cnts'
WHERE centre='cmts';
END;
/

Script Output x
Task completed in 0.023 seconds

anonymous block completed

```

11) REGLES DE PASSAGE DU DIAGRAMME DE CLASSE AU SH2MA RELATIONNEL

diagramme de classe

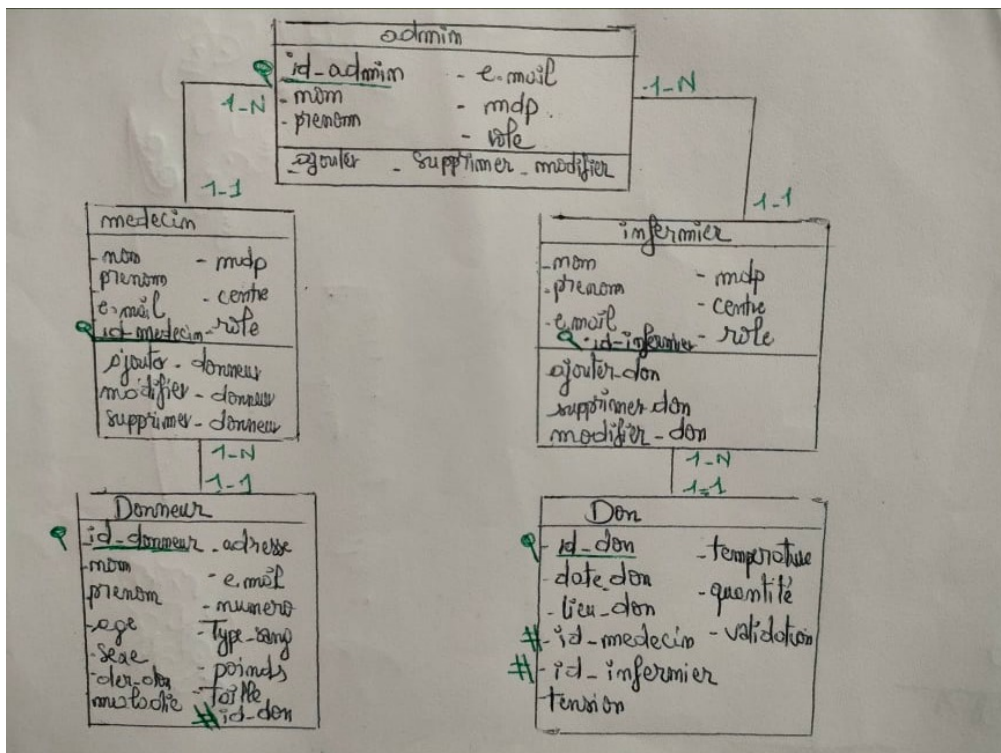
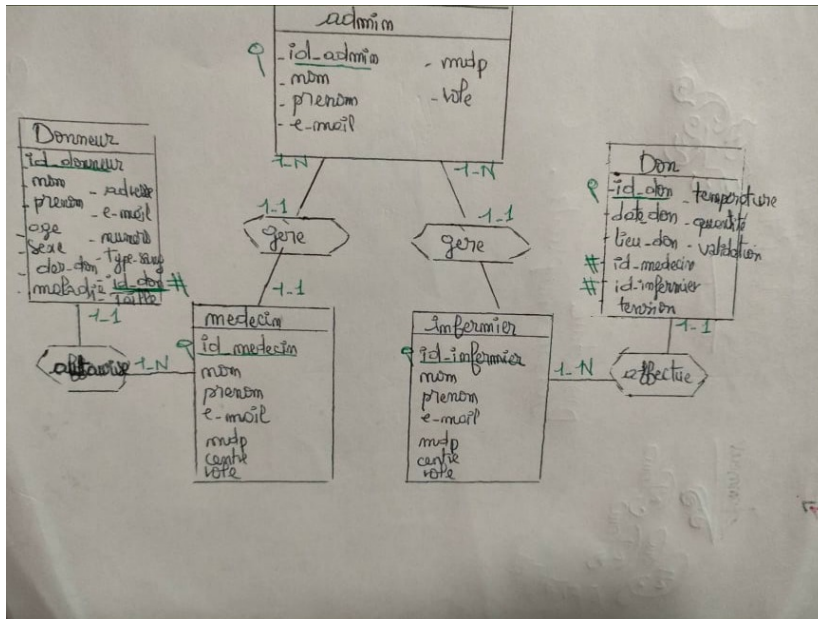


schéma relationnel



11) MES SCRIPT DE CRÉATION DES TABLES

```
--
-- Base de données : `sgcs`
--
-- Structure de la table `admin`
--

CREATE TABLE `admin` (
  `id_admin` double NOT NULL,
  `nom` varchar(20) NOT NULL,
  `prenom` varchar(20) NOT NULL,
  `email` varchar(100) NOT NULL,
  `mdp` varchar(40) NOT NULL,
  `role` varchar(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Déchargement des données de la table `admin`
--
```

```

INSERT INTO `admin` (`id_admin`, `nom`, `prenom`, `email`, `mdp`,
`role`) VALUES
(11438389, 'imen', 'fredj', 'imen.fredj@esen.tn', 'imen', 'admin'),
(14450427, 'nada', 'arfaoui', 'nada.arfaoui@esen.tn', 'nada', 'admin');

```

```

-- -----

```

```

--
-- Structure de la table `don`
--

```

```

CREATE TABLE `don` (
  `id_don` double NOT NULL,
  `date_don` date NOT NULL,
  `lieu_don` varchar(30) NOT NULL,
  `id_medecin` double NOT NULL,
  `id_infirmier` double NOT NULL,
  `tension` int(11) NOT NULL,
  `temperature` int(11) NOT NULL,
  `quantite` int(11) NOT NULL,
  `validation` tinyint(1) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

```

```

--
-- Déchargement des données de la table `don`
--

```

```

INSERT INTO `don` (`id_don`, `date_don`, `lieu_don`, `id_medecin`,
`id_infirmier`, `tension`, `temperature`, `quantite`, `validation`)
VALUES
(100, '2026-04-22', 'manouba', 11567820, 11432578, 121, 37, 450, 1),
(101, '2027-04-22', 'manouba', 52480489, 14520369, 121, 38, 425, 1),
(102, '2027-04-22', 'tunis', 11446630, 11432578, 119, 38, 425, 1),
(103, '2002-02-22', 'manouba', 33235780, 25366660, 121, 38, 450, 1),
(104, '2015-01-22', 'bardo', 11446630, 14520369, 120, 37, 425, 1),
(105, '2027-04-22', 'ariana', 33235780, 35780922, 121, 38, 0, 0);

```

```

-- -----

```

```

--
-- Structure de la table `donneur`
--

```

```

CREATE TABLE `donneur` (
  `id_donneur` double NOT NULL,
  `nom` varchar(20) NOT NULL,
  `prenom` varchar(20) NOT NULL,
  `age` int(11) NOT NULL,
  `sexe` varchar(10) NOT NULL,
  `adresse` varchar(50) NOT NULL,
  `email` varchar(100) NOT NULL,
  `numero` double NOT NULL,
  `type_sang` varchar(3) NOT NULL,
  `poids` int(11) NOT NULL,
  `taille` int(11) NOT NULL,
  `dernier_don` date NOT NULL,
  `maladie` varchar(50) NOT NULL,
  `id_don` double NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Déchargement des données de la table `donneur`
--

INSERT INTO `donneur` (`id_donneur`, `nom`, `prenom`, `age`, `sexe`,
`adresse`, `email`, `numero`, `type_sang`, `poids`, `taille`,
`dernier_don`, `maladie`, `id_don`) VALUES
(14050231, 'ftouhi', 'salma', 38, 'f', 'tunis',
'salma.ftouhi@gmail.com', 28478524, 'B-', 85, 170, '2018-11-10',
'saine', 104),
(14425876, 'madani', 'ahmed', 43, 'm', 'ariana',
'ahmed.madani@gmail.com', 99124568, 'O-', 86, 161, '2021-02-12',
'tension', 105),
(14450111, 'marnissi', 'ghofrane', 20, 'f', 'manouba',
'ghofrane.marnissi@gmail.com', 28557132, 'A+', 61, 155, '2020-01-12',
'saine', 103),
(14450123, 'riahi', 'amine', 35, 'm', 'tunis', 'amine.riahi@gmail.com',
98954157, 'O+', 60, 160, '2020-01-22', 'sain', 102),
(14450421, 'hamdi', 'olfa', 20, 'f', 'ben arouss',
'olfa.hamdi@gmail.com', 99124578, 'A+', 55, 155, '2022-04-26', 'saine',
100),
(14450422, 'jouini', 'hedi', 45, 'm', 'ouina', 'hedi.jouini@gmail.com',
99124478, 'O+', 72, 166, '2020-03-20', 'sain', 101);

--

```



```
--
-- Structure de la table `infirmier`
--

CREATE TABLE `infirmier` (
  `id_infirmier` double NOT NULL,
  `nom` varchar(20) NOT NULL,
  `prenom` varchar(20) NOT NULL,
  `email` varchar(100) NOT NULL,
  `mdp` varchar(40) NOT NULL,
  `centre` varchar(100) NOT NULL,
  `role` varchar(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
-- Déchargement des données de la table `infirmier`
--

INSERT INTO `infirmier` (`id_infirmier`, `nom`, `prenom`, `email`,
`mdp`, `centre`, `role`) VALUES
(11432578, 'amira', 'hamzaoui', '', '', 'cmts', 'infirmier'),
(14520369, 'arfaoui', 'asma', '', '', 'banque du sang', 'infirmier'),
(25366660, 'hamadi', 'mansour', '', '', 'cnts', 'infirmier'),
(26850413, 'gharbi', 'aicha', '', '', 'cmts', 'infirmier'),
(35780922, 'rjab', 'chaabene', '', '', 'cnts', 'infirmier');

-----

--
-- Structure de la table `medecin`
--

CREATE TABLE `medecin` (
  `id_medecin` double NOT NULL,
  `nom` varchar(20) NOT NULL,
  `prenom` varchar(20) NOT NULL,
  `email` varchar(100) NOT NULL,
  `mdp` varchar(40) NOT NULL,
  `centre` varchar(50) NOT NULL,
  `role` varchar(10) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

--
```

```
-- Déchargement des données de la table `medecin`
--

INSERT INTO `medecin` (`id_medecin`, `nom`, `prenom`, `email`, `mdp`,
`centre`, `role`) VALUES
(11436367, 'labyedh', 'slimen', '', '', 'cnts', 'medecin'),
(11446630, 'zarrouk', 'zaineb', '', '', 'cnts', 'medecin'),
(11567820, 'fracj', 'dyna', '', '', 'cmts', 'medecin'),
(33235780, 'charfi', 'nour', '', '', 'banque du sang', 'medecin'),
(52480489, 'ben arouss', 'touhemi', '', '', 'cmts', 'medecin');

-----
```